A Distributed Cloud-based Cyberinfrastructure Framework for Integrated Bridge Monitoring

Seongwoon Jeong*[a], Rui Hou[b], Jerome P. Lynch[b], Hoon Sohn[c], Kincho H. Law[a]
[a]Dept. of Civil & Environmental Engineering, Stanford University, Stanford, CA, USA 94305;
[b]Dept. of Civil & Environmental Engineering, University of Michigan, Ann Arbor, MI, USA 48109;
[c]Dept. of Civil & Environmental Engineering, KAIST, Daejeon 305-701, Republic of Korea;

ABSTRACT

This paper describes a cloud-based cyberinfrastructure framework for the management of the diverse data involved in bridge monitoring. Bridge monitoring involves various hardware systems, software tools and laborious activities that include, for examples, a structural health monitoring (SHM), sensor network, engineering analysis programs and visual inspection. Very often, these monitoring systems, tools and activities are not coordinated, and the collected information are not shared. A well-designed integrated data management framework can support the effective use of the data and, thereby, enhance bridge management and maintenance operations. The cloud-based cyberinfrastructure framework presented herein is designed to manage not only sensor measurement data acquired from the SHM system, but also other relevant information, such as bridge engineering model and traffic videos, in an integrated manner. For the scalability and flexibility, cloud computing services and distributed database systems are employed. The information stored can be accessed through standard web interfaces. For demonstration, the cyberinfrastructure system is implemented for the monitoring of the bridges located along the I-275 Corridor in the state of Michigan.

Keywords: Bridge monitoring, bridge management, cloud computing, distributed database

1. INTRODUCTION

Advances in sensor network technologies have led to instrumentations of sensors for monitoring the integrity of civil infrastructures. Sensor measurements have been utilized to enhance bridge operation and maintenance and public safety. For example, sensor data can be used to discover the onset of damage of a structure [1] as well as to assess the long-term performance of a structure [2, 3]. Most research efforts in structural health monitoring (SHM) focus on the development of sensor technologies and analytical damage detection methods. Comparatively, very few efforts have been devoted on the effective management of the data. Sensor measurement data is being stored and managed separate from bridge management operations. Decision making in bridge management involves not only the sensor measurements, but also other types of information such as visual inspection reports, traffic data and engineering analysis and assessments. The islands of data require that bridge managers retrieve the data using different tools and different interfaces, and then manually combines the different types of information for a specific task. A data infrastructure framework that allows seamless integration of the diverse sets of data can potentially enhance bridge management and maintenance operations.

Several efforts on the development of data management framework for the structural monitoring applications have been reported. For example, a self-managing software framework that supports sensor data management, remote data access and autonomous data analysis was developed for the monitoring of a wind turbine structure [4]. The research team at the University of Michigan developed a scalable data infrastructure platform, called SenStore, for the management of sensor measurements and engineering data and for supporting bridge monitoring and engineering analysis [5]. As the amount of measurement data increases and the variety of the collected information expands, a scalable and flexible data management framework becomes important. In the Internet of Things (IoT) domain, data infrastructure is among one of most critical issues as the amount of data collected by a wide variety of sensors has grown exponentially and will continue to do so [6]. State-of-the-art computing and information technologies are deployed for IoT applications. One of the enablers of IoT is cloud computing. Cloud computing provides cost-effective and scalable web-based computational resources, which are beneficial for handling very large-scale data sets collected from many different sources [7]. This study explores the use of cloud computing on infrastructure monitoring. Specifically, a cloud-based data management

* e-mail: swjeong3@stanford.edu

Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2017,
edited by Jerome P. Lynch, Proc. of SPIE Vol. 10168, 101682W · © 2017 SPIE
CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2270716

framework is deployed to efficiently manage the sensor data along with the domain specific information for bridge monitoring.

The cloud-based cyberinfrastructure framework for bridge monitoring is designed to manage in an integrated manner not only the sensor measurement data, but also other information involved in the decision-making processes of bridge management. The framework manages real-time sensor data collected from the sensor network, real-time traffic video images from traffic monitoring system, and the bridge information model data. The cyberinfrastructure, which serves as a data hub, offers easy access to the data and applications residing in the cloud. A distributed cloud-based data infrastructure is implemented to support the interoperability within and across cloud platforms. To facilitate information sharing and service invocation and integration, the cyberinfrastructure exposes its resources via web-APIs that adhere to *de facto* web service standards, namely the representational state transfer (REST) style architecture [8]. For data management in cloud environment, in which computing resources can be dynamically scaled, a highly scalable NoSQL-based distributed database system is employed. For testing and validation, the cyberinfrastructure framework has been implemented for the monitoring of the bridges along the I-275 Corridor in the state of Michigan.

2. CYBERINFRASTRUCTURE FRAMEWORK

Figure 1 depicts the conceptual framework of the cloud-based cyberinfrastructure. The cloud-based cyberinfrastructure offers the integrated management of heterogeneous information involved in bridge monitoring applications. The real-time sensor data acquired from the sensor network on bridge structures is streamed to the database system installed on the cloud. The data about the bridge, such as its geometry and finite element model, as well as sensor information and inspection reports are also stored in the cloud database. Additionally, a wide variety of data collected from external public sources, such as traffic monitoring cameras and weather stations, that can provide meaningful information for bridge management are collected and stored. The integrated database system manages a diverse set of heterogeneous data that encompasses time-series data, object-oriented engineering data, images and text documents. Standard web service APIs are employed for building and linking applications, such as web/mobile apps, engineering tools (e.g., CAD and structural analysis software tools) and data analytics modules, and for allowing efficient retrieval of the data stored in the database. The cyberinfrastructure framework is designed in such a way that the cloud services implemented in the cloud platform can be integrated with other cloud environments, thereby, the architectural framework can conceptually support hybrid cloud systems and enable the construction of a system of systems.

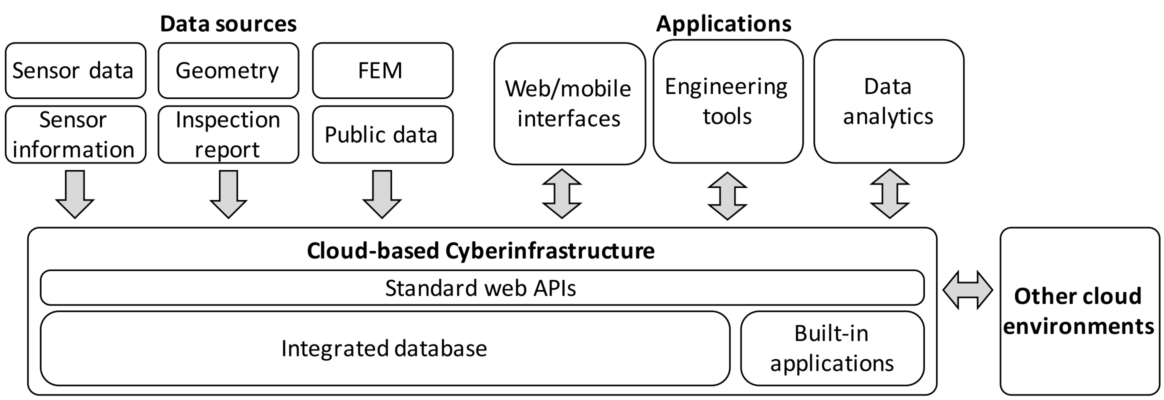


Figure 1. Conceptual framework of the cloud-based cyberinfrastructure for bridge monitoring

The keys to build an integrated database management infrastructure include (1) a flexible and scalable database system, (2) an integrated data schema and ontology and (3) consistent and interoperable interfaces. The database management system needs to be able to support flexible data structure for handling heterogeneous data sets and scalable to manage massive volume of data. An integrated data schema, which defines the relations and links between data entities, is essential to help relate and combine heterogeneous data sets. Standard interfaces are needed to allow easy access to the data by human user and software applications. The adoption of standard interfaces can also facilitate interoperability within a cyberinfrastructure environment and across the different cloud environments by enabling the invocation and integration of services (e.g., data retrieval and service execution).

2.1 Cloud-based cyberinfrastructure architecture

The cloud-based cyberinfrastructure is built upon the client-server model, where the server provides data and applications as services that clients can use. Serving as a server system, the cyberinfrastructure system delivers the data and applications through standard web services. Figure 2 shows the architecture of the cloud-based cyberinfrastructure. The virtual machines (VMs) provided by the cloud vendors serve as the computing infrastructure. The database system, which runs on the operating system (OS) provided by the VMs, offers query languages and APIs that other systems can use to store and query data. The web server hosts the web services, which can be invoked by client users or applications via the Hypertext Transfer Protocol (HTTP) and delivers data and application results in standard web-based syntax.

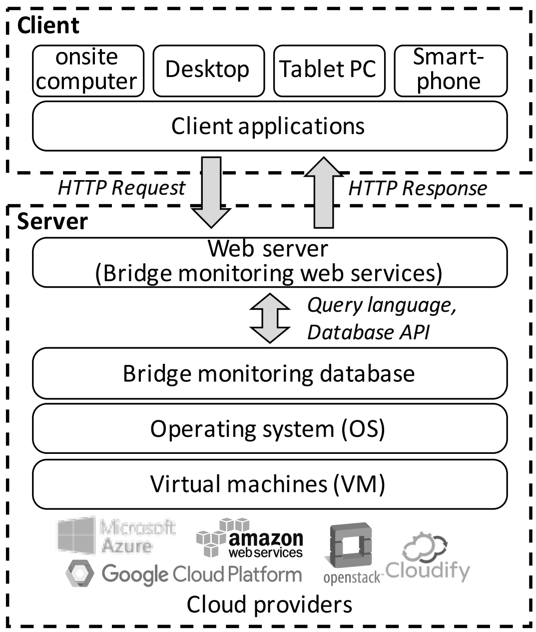


Figure 2. Client-server architecture for cyberinfrastructure

The cyberinfrastructure that leverages cloud computing enables flexible and scalable system deployment. For bridge monitoring application, the size of sensor data would grow and the demand (e.g., periodically executing data analysis) on computing resources varies. The flexibility and scalability features of cloud computing can facilitate the optimized use of computing resources.

2.2 Distributed database

Database systems that can take advantage of the features specific to cloud environment are key to maximize the benefits of cloud computing. Traditional relational database (RDB) systems do not fully embrace many of the cloud features such as partitioning tolerance and high availability [9]. NoSQL (Not-only-SQL) database systems have emerged as a viable alternative [10]. Many open-source NoSQL database systems, each of which offers different performance advantages, are now available. In the current prototype implementation, a column-oriented NoSQL database system, namely Apache Cassandra database (http://cassandra.apache.org/), is selected for its scalability. Cassandra database is built upon a "ring" architecture as illustrated in Figure 3. The ring architecture consists of several nodes, each of which is a database instance running on a computer. The nodes communicate with each other via the network to distribute the data across the cluster and, at the same time, to maintain consistency of a database. Unlike a master-slave architecture, in the ring architecture, an identical role is assigned to the nodes so that failures in a few nodes do not lead the failure of an entire database.

The partitioning engine of the Cassandra database plays a key role in distributing the data sets across the nodes in the database cluster and in rebalancing the loads among the nodes when scaling the cluster. The data sets are replicated across the nodes to ensure high availability and fault-tolerance. For example, as shown in Figure 3, the incoming data set *R* is partitioned into two parts *r1* and *r2* which are replicated three times, and distributed to six different nodes, so that the failure of any single node (say node *N3*) does not prohibit the write and read operations.

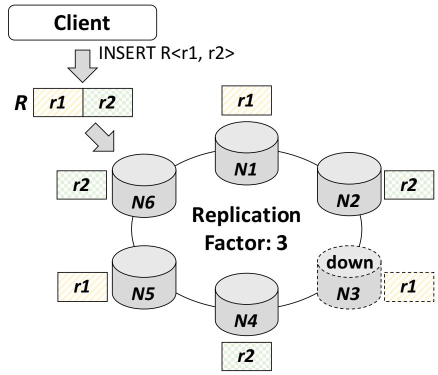


Figure 3. The ring architecture of the Cassandra database

The column-oriented data model of the Cassandra database consists of "keyspace," "column family," "row" and "column," which are analogous to "database," "table," "tuple" and "attribute", respectively, of RDB. One major difference between the column-oriented model and the tabular relational model is that the column-oriented model can have rows that contain different sets of columns, while the relational model has the same schema for every tuple. The flexibility of having varying numbers of column (attributes) makes the column-oriented model more suitable in handling heterogeneous data created in bridge monitoring. Figure 4 shows an excerpt of the database schema representing the various information stored for bridge monitoring applications. The data schema includes column families for storing and managing sensor measurement data, sensor information, bridge geometry, finite element model and traffic video images. Data links between relevant data entities (e.g., sensor and its location in a finite element model) are defined to facilitate integrated use of different types of data from different sources.

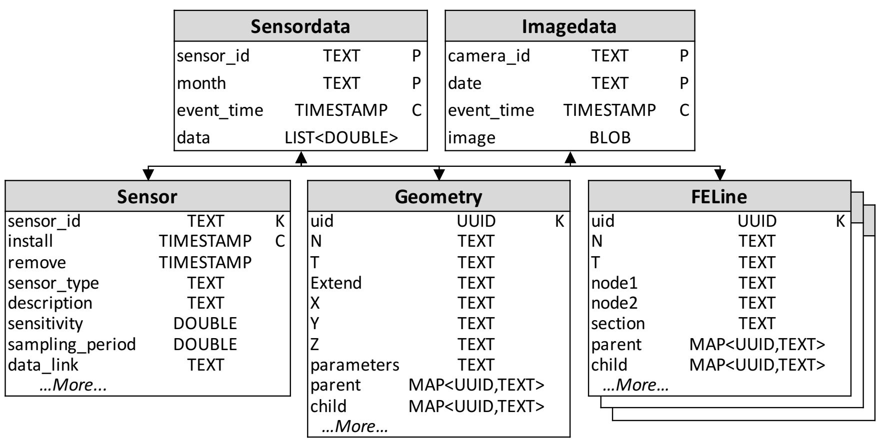


Figure 4. Data schema definition

2.3 Web server

The web server component is designed to provide consistent and interoperable interfaces to the cloud-based cyberinfrastructure. The interfaces are offered as web services, which support the "interoperable machine-to-machine interaction over a network" [11]. The web services are designed to follow the *de facto* REST architecture style to facilitate interoperability. The RESTful web services (i.e., web services that follow REST architecture style) use the basic HTTP methods, including PUT, GET, POST and DELETE, as unified interfaces by which clients can manipulate resources on the server. The resources are identified using the Uniform Resource Identifiers (URIs) through which a client user or an application can send requests. The messages (e.g., a response returning to a client) in REST architecture style are written in self-descriptive formats, such as eXtensible Markup Language (XML) or JavaScript Object Notation (JSON). Since the RESTful web services are platform-neutral, the web services can be invoked by different devices and operating systems including desktop computers, laptop computers, tablet PCs and smartphones. Furthermore, the standard web services can be invoked within and across the cloud environment, thereby facilitating the integration of web services residing in any cloud platform. In the current implementation of the web server, we have developed a number of RESTful web services for the retrieval of different types of data involved in the bridge monitoring applications. Table 1 summarizes the web services that have been developed in the current prototype cyberinfrastructure system. For developing the scalable and high-performing web server, we employ Node.js (https://nodejs.org/), a non-blocking web server runtime environment. In addition, we employ NGINX (https://www.nginx.com/) as a load balancer to enable the distribution of workloads over multiple web servers.

Table 1. RESTful web services currently implemented on the web server

URI	HTTP method	Parameters	Returning content type	Description
/sensor	GET	sensortype, install, remove	application/json	Retrieving the list of sensors.
/sensor/{id}	GET	property, install, remove	application/json	Retrieving sensor information of the sensor whose ID is {id}.
/daqevent	GET	event_time_begin, event_time_end	application/json	Retrieving the list of data acquisition events.
/sensordata/{id}	GET	month, event_time_begin, event_time_end	application/json	Retrieving sensor data collected by the sensor whose ID is {id}.
/imagedata/{id}	GET	date, event_time_begin, event_time_end	application/json	Retrieving image data collected by the camera whose ID is {id}.
/femodel/{id}	GET	format	application/xml, application/vnd.ms-excel	Retrieving the finite element model of the bridge whose ID is {id} by invoking data parser applications. Client can choose file type from XML and XLSX (Microsoft Excel).
/geometricmodel/{id}	GET	-	application/xml	Retrieving the geometric model of the bridge whose ID is {id}.

3. PROTOTYPE IMPLEMENTATION

This section describes the prototype implementation of the cloud-based cyberinfrastructure for bridge monitoring applications. The cyberinfrastructure is designed and implemented for the bridge monitoring system installed on the Telegraph Road Bridge (Figure 5(a)) and the Newburg Road Bridge (Figure 5(b)), which are located along the I-275 corridor in Michigan. The two bridges are equipped with sensors that measure acceleration, strain and temperature of the bridge structures. Additionally, video cameras are installed by the public agency for traffic monitoring.

(a) Telegraph Road Bridge (b) Newburg Road Bridge

Figure 5. Testbed bridges on I-275 Corridor

3.1 Cloud-based cyberinfrastructure implementation

For the management of the data collected from the two bridges, we employ a number of Linux-based VMs (Ubuntu Linux Server 16.04.1 LTS) from the Microsoft Azure public cloud platform. The provisioned VMs can be remotely accessed via a network protocol, such as Secure Shell (SSH) and Secure Copy Protocol (SCP), to deploy the platform, to install applications and to transmit files. We use multiple VMs to build a distributed database and to host the web server apart from the database servers to distribute the workloads. Specifically, we use five virtual machines to construct the distributed Cassandra database and a single virtual machine to build the web server using Node.js and NGINX. The configurations and the roles of the VMs are shown in Table 2.

Table 2. Specification and role of cloud servers in the implementation

No.	VM model name	OS	CPU	Memory	Role
1	Azure Standard_A2m_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores	16 GB	Database node
2	Azure Standard_A2m_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores	16 GB	Database node
3	Azure Standard_A2m_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores	16 GB	Database node
4	Azure Standard_A2m_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores	16 GB	Database node
5	Azure Standard_A2m_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores	16 GB	Database node
6	Azure Standard_DS2_v2	Ubuntu Linux Server 16.04.1 LTS	2 cores (high-performing CPU)	7 GB	Web server, load balancer

The cyberinfrastructure receives sensor measurement data from the sensor networks via the microcomputers installed at the bridge sites. We also upload the sensor information, geometric model and the finite element model of the Telegraph Road Bridge to the cyberinfrastructure. Furthermore, we develop scripts to automatically collect public data from external data sources. For example, we develop a traffic video image data collector to store the traffic images so that the number and the types of vehicles passing the bridges can be automatically analyzed. Containing the list of the Uniform Resource Locators (URLs) of data sources, the data collector repeatedly accesses the data sources via the Internet, parses the resources to locate the image data stream, and uploads the image data to the cloud-based cyberinfrastructure system.

The data managed by cyberinfrastructure can be accessed through web services. For example, Figure 6 shows the sensor information retrieved from the cyberinfrastructure by issuing an HTTP request. The request specifies the HTTP method type "GET" and the URI of the resources. The web server of the cyberinfrastructure processes the request and returns a response message that contains the requested data encoded in JSON.

GET: http://x.x.x.x/sensor/u07ch0


```
{
  "content":[
    {
      "sensor_id":"u07ch0",
      "install":"2013-10-01T00:00:00.000Z",
      "conversion_factor":0.0152587890625,
      "description":"Acceleration sensor installed on the Telegraph Road Bridge",
      "global_coordinate":"42.018389,-83.346720",
      "intended_application":"Acceleration",
      "local_coordinate":"102,-50",
      "output":"Acceleration",
      "output_uom":"mg",
      "position_description":"Girder",
      "sampling_rate":200,
      "sensitivity":"1 V/g",
      "sensor_type":"Accelerometer"
    }
  ], ... More data ...
}
```


Figure 6. Sensor information retrieved via a web service of the cyberinfrastructure

3.2 Integrated bridge monitoring interface

An integrated bridge monitoring user interface is developed to support access to the bridge monitoring data on the cloud platform through human-readable web pages. The user interface, implemented as a client application in the client-server architecture, enables the interaction between the web pages and the users, invokes web services and returns information to the users. The current design of the user interface supports the retrieval of various bridge monitoring information, as illustrated in Figure 7. For the retrieval of sensor information, a user can enter the query parameters, such as ID of sensor and sensor type. In addition, a user can retrieve sensor data and image data by specifying sensor ID (or camera ID) and the time duration. A user can also download bridge model by entering the name of the bridge and clicking the buttons that indicate the type of the model (e.g., geometric model in XML format, FE model in XML format and FE model in Microsoft Excel format). The downloaded models can be used by appropriate tools, such as OpenBrIM viewer (https://openbrim.org/) for 3-dimensional geometric model and CSI Bridge (https://www.csiamerica.com/products/csibridge) for bridge engineering model.

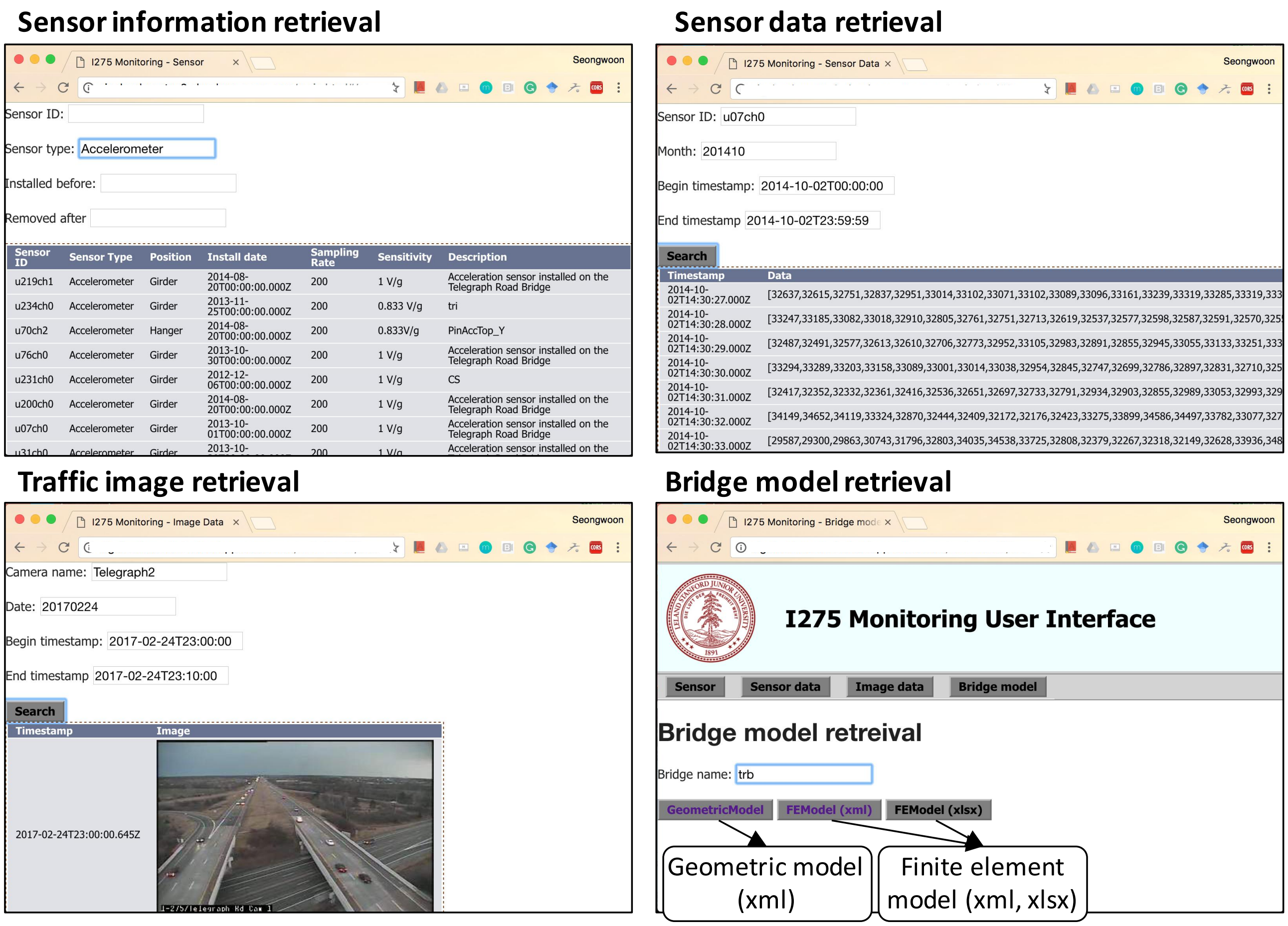
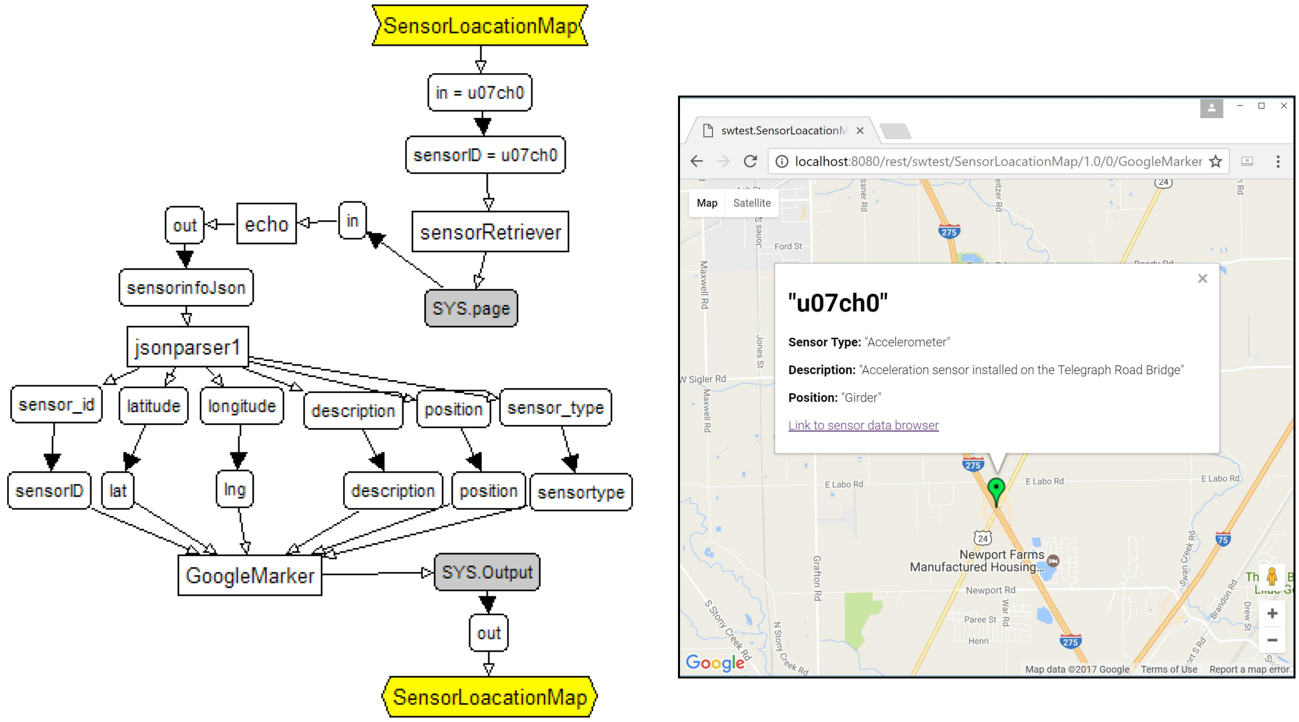


Figure 7. Screenshots of the preliminary web user interface

3.3 Service composition

The standardized web services offered by the cyberinfrastructure can be invoked and reused to compose new web services from a local computer or from another cloud platform. Different service composition methods have been suggested to support the integration of RESTful web services [12-14]. For demonstration, we build a simple application named SensorLocationMap that composes web services by using JOpera (http://www.jopera.org/), a visual web service composition tool. The SensorLocationMap reads the sensor ID as an input argument and returns a map on which the sensor's information and the data link to the sensor data repository are displayed. Figure 8(a) shows the application workflow (visualized by JOpera) that specifies a sequence of applications and the input and output of each application. The composition tool orchestrates the sensor information retrieval service (GET /sensor/:id) on the cyberinfrastructure system with other services, such as the Google Map API (https://developers.google.com/maps/). When executed, the application runs sensor information retrieval service with the input argument that specifies the target sensor ID. The web service returns a JSON snippet that describes the sensor information. The jsonparser1 module (written in JavaScript) then parses the snippet, extracts sensor information (e.g., sensor location, sensor type and description) and delivers the information to the GoogleMarker module that displays the information on the Google Map using the Google Map API. Figure 8(b) shows the sensor information and the location marker on the Google Map generated by the composite application. This example illustrates that the cyberinfrastructure's web services can be invoked and used by other services to compose new services.



(a) Application workflow visualized by JOpera (b) A running instance of application

Figure 8. Web service composition example

4. SUMMARY AND CONCLUSION

This paper describes the development of a cloud-based cyberinfrastructure for integrated information management for bridge monitoring. The cyberinfrastructure framework is designed to manage not only the real-time sensor measurement data, but also other relevant information, such as a bridge model, engineering model and traffic images. For scalability and flexibility, we use cloud computing services as the computing infrastructure. We employ distributed database system for efficient management of heterogeneous data sets. In addition, an integrated data schema is defined to relate and combine heterogeneous data involved in bridge monitoring. The cyberinfrastructure also offers standardized web services that can be invoked and integrated within and across cloud environments. For the validation of the framework, the cyberinfrastructure is implemented for the monitoring of the bridges along the I-275 Corridor in the state of Michigan. The results show that the cyberinfrastructure can serve as a data hub for the bridge monitoring and management through which users and applications can access different types of bridge monitoring information.

ACKNOWLEDGEMENT

This research is supported by a Grant No. 13SCIPA01 from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA). The research is also partially supported by a collaborative project funded by the US National Science Foundation (Grant No. ECCS-1446330 to Stanford University and Grant No. ECCS-1446521 to the University of Michigan). The authors thank the Michigan Department of Transportation (MDOT) for access to the Telegraph Road Bridge and the Newburg Road Bridge and for offering support during installation of the wireless monitoring system. Any opinions, findings, conclusions or recommendations expressed in this paper are solely those of the authors and do not necessarily reflect the views of NSF, MOLIT, MDOT, KAIA or any other organizations and collaborators.

REFERENCES

[1] Lim, H. J., Sohn, H. and Liu, P., “Binding conditions for nonlinear ultrasonic generation unifying wave propagation and vibration,” *Applied Physics Letters*, 104(21), 214103 (2014).

[2] Cross, E. J., Koo, K. Y., Brownjohn, J. M. W. and Worden, K., “Long-term monitoring and data analysis of the Tamar Bridge,” *Mechanical Systems and Signal Processing*, 35(1), 16-34 (2013).

[3] O'Connor, S. M., Zhang, Y., Lynch, J., Ettouney, M. and van der Linden, G., “Automated analysis of long-term bridge behavior and health using a cyber-enabled wireless monitoring system,” *Proceedings of the SPIE Smart Structures/NDE Conference*, Art. No. 90630Y (2014).

[4] Smarsly, K., Law, K. H. and Hartmann, D., “Multiagent-based collaborative framework for a self-managing structural health monitoring system,” *Journal of computing in civil engineering*, 26(1), 76-89 (2012).

[5] Zhang, Y., O’Connor, S. M., van der Linden, G., Prakash, A. and Lynch, J. P., “SenStore: A Scalable Cyberinfrastructure Platform for Implementation of Data-to-Decision Frameworks for Infrastructure Health Management,” *Journal of Computing in Civil Engineering*, 04016012 (2016).

[6] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.H., “Big data: The next frontier for innovation, competition, and productivity,” McKinsey Global Institute (2011).

[7] Zaslavsky, A., Perera, C. and Georgakopoulos, D., “Sensing as a service and big data,” *In International Conference on Advances in Cloud Computing* (ACC-2012), 21–29 (2012).

[8] Fielding, R. T., “Architectural styles and the design of network-based software architectures,” Doctoral dissertation, University of California, Irvine (2000).

[9] Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N. and Helland, P., “The end of an architectural era (it's time for a complete rewrite),” *Proceedings of the 33rd International Conference on Very large data bases*, 1150–1160 (2007).

[10] Grolinger, K., Higashino, W. A., Tiwari, A. and Capretz, M. A., “Data management in cloud environments: NoSQL and NewSQL data stores,” *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1), 22–41 (2013).

[11] Haas, H. and Brown, A., “Web Services Glossary,” W3C Working Group Note, 2004, <https://www.w3.org/TR/ws-gloss/> (accessed on 30 January 2017).

[12] Rosenberg, F., Curbera, F., Duftler, M. J. and Khalaf, R., “Composing RESTful Services and Collaborative Workflows: A Lightweight Approach,” *In IEEE Internet Computing*, 12(5), 24–31 (2008).

[13] Zhao, H. and Doshi, P., “Towards automated restful web service composition,” *2009 IEEE International Conference on Web Services*, 189-196 (2009).

[14] Pautasso, C., “BPEL for REST,” *In International Conference on Business Process Management*, 278–293 (2009).